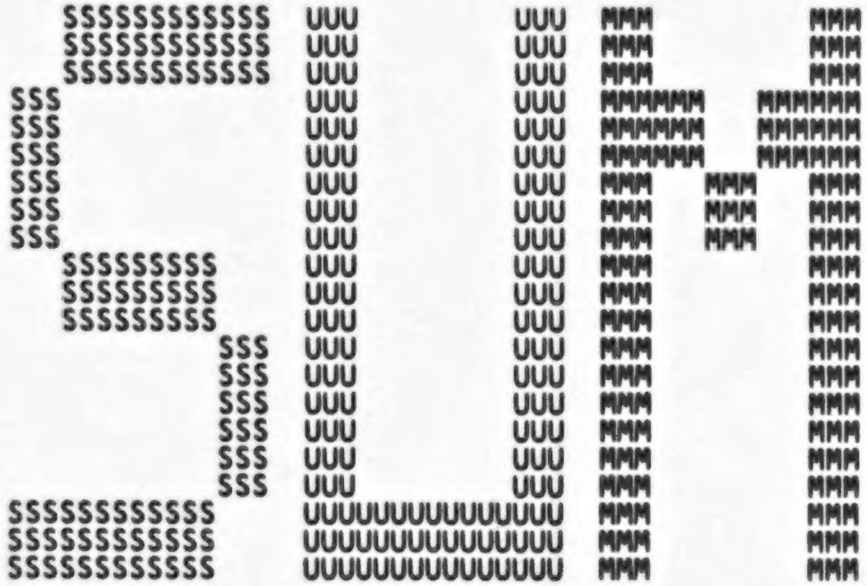
SSSSSSSSSSS UUU UUU MMM MMM
SSSSSSSSSSS UUU UUU MMM MMM
SSSSSSSSSSS UUU UUU MMM MMM
SSS UUU UUU MMMMMM MMMMMM
SSS UUU UUU MMMMMM MMMMMM
SSS UUU UUU MMMMMM MMMMMM
SSS UUU UUU MMM MMM MMM
SSS UUU UUU MMM MMM MMM
SSS UUU UUU MMM MMM MMM
SSSSSSSSS UUU UUU MMM MMM
SSSSSSSSS UUU UUU MMM MMM
SSSSSSSSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSSSSSSSSSS UUUUUUUUUUUUUUUU MMM MMM
SSSSSSSSSSS UUUUUUUUUUUUUUUU MMM MMM
SSSSSSSSSSS UUUUUUUUUUUUUUUU MMM MMM

FILEID**SUMFILES

```
  SSSSSSSS  UU      UU  MM      MM  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
  SSSSSSSS  UU      UU  MM      MM  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
SS          UU      UU  MMMM  MMMM  FF              II      LL          EE          SS
SS          UU      UU  MMMM  MMMM  FF              II      LL          EE          SS
SS          UU      UU  MM  MM  MM  FF              II      LL          EE          SS
SS          UU      UU  MM  MM  MM  FF              II      LL          EE          SS
  SSSSSS    UU      UU  MM      MM  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
  SSSSSS    UU      UU  MM      MM  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
        SS  UU      UU  MM      MM  FF              II      LL          EE                  SS
        SS  UU      UU  MM      MM  FF              II      LL          EE                  SS
        SS  UU      UU  MM      MM  FF              II      LL          EE                  SS      ....
        SS  UU      UU  MM      MM  FF              II      LL          EE                  SS      ....
SSSSSSSS    UUUUUUUUUU  MM      MM  FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS        ....
SSSSSSSS    UUUUUUUUUU  MM      MM  FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
                     0000     1 ;
                     0000     2 ; Version:      'V04-000'
                     0000     3 ;
                     0000     4 ;****************************************************************************
                     0000     5 ;*                                                                          *
                     0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
                     0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
                     0000     8 ;*  ALL RIGHTS RESERVED.                                                    *
                     0000     9 ;*                                                                          *
                     0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
                     0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
                     0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
                     0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
                     0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
                     0000    15 ;*  TRANSFERRED.                                                            *
                     0000    16 ;*                                                                          *
                     0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
                     0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
                     0000    19 ;*  CORPORATION.                                                            *
                     0000    20 ;*                                                                          *
                     0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
                     0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
                     0000    23 ;*                                                                          *
                     0000    24 ;*                                                                          *
                     0000    25 ;****************************************************************************
                     0000    26 ;
                     0000    27 ;
                     0000    28 ; Assembly parameters
                     0000    29 ;
00000200             0000    30         BUF_SIZE = 512                  ; Size in bytes of slipr input buffers
00000084             0000    31         CMD_SIZE = 132                  ; Size of input command line
                     0000    32 ;
                     0000    33         $NAMDEF
                     0000    34         $RABDEF
                     0000    35         $FABDEF
                     0000    36         $CLIDEF
                     0000    37 ;
                     0000    38 ; Edit node offsets
                     0000    39 ;
00000000             0000    40         ED$L_FWD   = 0                  ; Forward pointer
00000004             0000    41         ED$L_BWD   = 4                  ; Backword pointer
00000008             0000    42         ED$W_LOC1  = 8                  ; Locator 1
0000000A             0000    43         ED$W_LOC2  = 10                 ; Locator 2
0000000C             0000    44         ED$W_LINES = 12                 ; Insert lines
0000000E             0000    45         ED$W_RFA   = 14                 ; Record file address (3 words)
00000014             0000    46         ED$L_FILE  = 20                 ; File node pointer
00000018             0000    47         ED$B_FLAGS = 24                 ; Flags
00000019             0000    48         ED$B_FILENO= 25                 ; File number
                     0000    49 ;
0000001A             0000    50         ED$K_BLN   = 26
                     0000    51 ;
                     0000    52 ;
                     0000    53 ; File node offsets
                     0000    54 ;
00000000             0000    55         SLP$L_FWD  = 0                  ; Forward pointer
00000004             0000    56         SLP$L_BWD  = 4                  ; Backward pointer
00000008             0000    57         SLP$W_LOC1 = 8                  ; Locator-1
```

```
0000000A  0000     58          SLP$W_LOC2 = 10                 ; Locator-2
0000000C  0000     59          SLP$B_FLAGS= 12                 ; Flags
0000000D  0000     60          SLP$B_FILENO = 13               ; File priority
0000000E  0000     61          SLP$W_DOT  = 14                 ; Dot value
00000010  0000     62          SLP$Q_AUDDS= 16                 ; Audit string desciptor
00000018  0000     63          SLP$T_AUDST= 24                 ; Audit string
00000028  0000     64          SLP$Q_AUCDS= 40                 ; Current audit string descriptor
00000030  0000     65          SLP$T_AUCST= 48                 ; Current audit string
00000040  0000     66          SLP$Q_CMNT = 64                 ; Comment descriptor
00000048  0000     67          SLP$T_NAM  = 72                 ; NAM block
          0000     68 ;
000000A8  0000     69          SLP$K_BLN  = SLP$T_NAM + NAM$K_BLN
          0000     70 ;
          0000     71 ;
          0000     72 ; Macro to print error message
          0000     73 ;
          0000     74          .MACRO  ERRMSG  NAME,LIST
          0000     75          $$ = 0
          0000     76          .IRP    L,<LIST>
          0000     77          PUSHL   L
          0000     78          $$=$$+1
          0000     79          .ENDR
          0000     80          PUSHL   #$$
          0000     81          MOVL    #MER$_'NAME',R0
          0000     82          PUSHL   R0
          0000     83          CALLS   #$$+2,G^LIB$SIGNAL
          0000     84          .ENDM   ERRMSG
```

```
                        1         .TITLE  SUMFILES
                        2         .IDENT  /V04-000/
                        3
                        4 ;
                        5 ;*****************************************************************************
                        6 ;*                                                                           *
                        7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
                        8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
                        9 ;*  ALL RIGHTS RESERVED.                                                     *
                       10 ;*                                                                           *
                       11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
                       12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
                       13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
                       14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
                       15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
                       16 ;*  TRANSFERRED.                                                             *
                       17 ;*                                                                           *
                       18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
                       19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
                       20 ;*  CORPORATION.                                                             *
                       21 ;*                                                                           *
                       22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
                       23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
                       24 ;*                                                                           *
                       25 ;*                                                                           *
                       26 ;*****************************************************************************
                       27 ;
                       28 ; Procedure to prompt user to supply a list of input files
                       29 ; and a single output file.  At least one input file must be
                       30 ; supplied.  The procedure will continue to prompt for input files
                       31 ; until at least one is supplied.  The single output file
                       32 ; is optional
                       33
                       34         $NAMDEF
                       35         $FABDEF
                       36 ;
                       37 ;
                       38 ;
                 00000000 39         .PSECT  $CODE,EXE,NOWRT
                       40
                       41 GET_FILES::
             0000      42         .WORD   0
      6D 0000'CF   0002 43         MOVAL   W^GET_HANDLER,(FP)      ; Set condition handler
                       44 10$:
0030'CF 0001'CF   0007 45         MOVAL   W^PROMPT_INPUT+1, -     ; Set up read prompt string
                       46                 W^CMD_INPUT_RAB+RAB$L_PBF
0034'CF 0000'CF   000E 47         MOVB    W^PROMPT_INPUT, -
                       48                 W^CMD_INPUT_RAB+RAB$B_PSZ
          1C   0015 49         BSB     INPUT_FILES             ; Get input files
       18 50   0017 50         BLBC    R0,20$                  ; If any errors start again
          5B   001A 51         TSTL    R11                     ; If zero input files given reprompt
          E9   001C 52         BEQL    10$
0030'CF 0001'CF   001E 53         MOVAL   W^PROMPT_OUTPUT+1, -    ; Set up 'Output' prompt string
                       54                 W^CMD_INPUT_RAB+RAB$L_PBF
0034'CF 0000'CF   0025 55         MOVB    W^PROMPT_OUTPUT, -
                       56                 W^CMD_INPUT_RAB+RAB$B_PSZ
        0167   002C 57         BSBW    OUTPUT_FILE             ; Get output file
```

```
00 50   E9   002F     58
                0032   59 20$:    BLBC    R0,20$                  ; If any errors start again
        04   0032     60          RET
```

```
                     0033     62          .SBTTL  INPUT_FILES
                     0033     63 ;
                     0033     64 ;
                     0033     65 ; Subroutine to get input files
                     0033     66 ;
                     0033     67 ; Inputs:
                     0033     68 ;       None
                     0033     69 ;
                     0033     70 ; Outputs:
                     0033     71 ;       R0 = Success/error status
                     0033     72 ;
                     0033     73 ;
                     0033     74 INPUT_FILES:
                5B   D4  0033     75          CLRL    R11                             ; Initialise input files count
0030'CF    0004'CF   D0  0035     76          MOVL    W^DEF_NAME+4,W^INPUT_FAB+FAB$L_DNA ; Set default file name
0035'CF    0000'CF   90  003C     77          MOVB    W^DEF_NAME,W^INPUT_FAB+FAB$B_DNS
                     0043     78 10$:
          56 0000'CF   DE  0043     79          MOVAL   W^INPUT_BUF,R6                  ; Set address to put file name string
               01AB   30  0048     80          BSBW    GETFILE                         ; Get next file
             29 50    E9  004B     81          BLBC    R0,40$                          ; Error if LBC
                57    95  004E     82          TSTB    R7                              ; Is file spec null (0 bytes)?
                1B    12  0050     83          BNEQ    30$                             ; No if NEQ
                5B    D5  0052     84          TSTL    R11                             ; Any files yet?
                03    12  0054     85          BNEQ    20$                             ; Yes if NEQ
             1E 58    E8  0056     86          BLBS    R8,40$                          ; End of list if LBS
                     0059     87 20$:
                     0059     88          ERRMSG  NULLFS                          ; Report error
                0A   11  006B     89          BRB     40$
                     006D     90 30$:
                5B   D6  006D     91          INCL    R11                             ; Increment file number
                07   10  006F     92          BSB     INPUT_SPEC                      ; Process spec
          03 50    E9  0071     93          BLBC    R0,40$                          ; Error if LBC
          CC 5B    E9  0074     94          BLBC    R8,10$                          ; More files if LBC
                     0077     95 40$:
                05   0077     96          RSB
```

```
                                0078     98          .SBTTL  INPUT_SPEC
                                0078     99 ;
                                0078    100 ;
                                0078    101 ;
                                0078    102 ; Inputs:
                                0078    103 ;       R6 = Address of file specification
                                0078    104 ;       R7 = Length of file specification
                                0078    105 ;
                                0078    106 ; Outputs:
                                0078    107 ;       R0 = Success/error status
                                0078    108 ; Subroutine to process input file spec
                                0078    109 ;
                                0078    110 ;
                                0078    111 INPUT_SPEC:
                  0000'CF   DF  0078    112         PUSHAL  W^VIRT_ADDR             ; Get slp file node
                  0000'CF   DF  007C    113         PUSHAL  W^SLP_SIZE
        00000000'GF    02   FB  0080    114         CALLS   #2,G^LIB$GET_VM
                       0B 50   E8  0087    115         BLBS    R0,10$                  ; OK if LBS
                          50   DD  008A    116         PUSHL   R0                      ; Signal error
        00000000'GF    01   FB  008C    117         CALLS   #1,G^LIB$SIGNAL
                       6B   11  0093    118         BRB     20$
                                0095    119 10$:
     00   0000'CF    00   2C  0095    120         MOVC5   #0,W^0,#0,L^SLP_SIZE, - ; Clear new memory
0000'DF    00000000'EF         009B
                                00A3    121                 @W^VIRT_ADDR
           52   0000'CF   D0  00A3    122         MOVL    W^VIRT_ADDR,R2          ; Set node pointer
             0D A2    5B  90  00A8    123         MOVB    R11,SLP$B_FILENO(R2)    ; Insert file priority number
         14 A2    18 A2   DE  00AC    124         MOVAL   SLP$T_AUDST(R2), -      ; Initialise audit string descriptor
                                00B1    125                 SLP$Q_AUDDS+4(R2)
         2C A2    30 A2   DE  00B1    126         MOVAL   SLP$T_AUCST(R2), -      ; Initialise audit string descriptor
                                00B6    127                 SLP$Q_AUCDS+4(R2)
                          04   BB  00B6    128         PUSHR   #^M<R2>                 ; Initialise with default string
           28 A2    0000'CF   B0  00B8    129         MOVW    W^DEF_AUDIT,SLP$Q_AUCDS(R2)
30 A2   0004'DF    0000'CF   28  00BE    130         MOVC3   W^DEF_AUDIT,@W^DEF_AUDIT+4, -
                                00C7    131                 SLP$T_AUCST(R2)
                          04   BA  00C7    132         POPR    #^M<R2>
                       53   52   D0  00C9    133         MOVL    R2,R3                   ; and NAM block pointer
                 53    00000048 8F   C0  00CC    134         ADDL    #SLP$T_NAM,R3
                    54    0000'CF   DE  00D3    135         MOVAL   W^INPUT_FAB,R4
                                00D8    136         $FAB_STORE FAB=R4, -            ; Set up FAB
                                00D8    137                 NAM = (R3), -
                                00D8    138                 FNA = (R6), FNS = R7
                                00E4    139         $NAM_STORE NAM = R3, -
                                00E4    140                 BID = #NAM$C_BID, -
                                00E4    141                 BLN = #NAM$C_BLN
                       13   10  00EC    142         BSB     PARSE_SPEC              ; Parse file spec
                    0F 50   E9  00EE    143         BLBC    R0,20$                  ; Error if LBC
                 0004'DF    62  0E  00F1    144         INSQUE  (R2),@W^FILE_NODES+4    ; Insert new file node
              30 A4    0C A3   D0  00F6    145         MOVL    NAM$L_ESA(R3),FAB$L_DNA(R4) ; Reset defaults
              35 A4    0B A3   90  00FB    146         MOVB    NAM$B_ESL(R3),FAB$B_DNS(R4)
                                0100    147 20$:
                                05  0100    148         RSB
```

```
                        0101   150          .SBTTL  PARSE_SPEC
                        0101   151 ;
                        0101   152 ;
                        0101   153 ; Subroutine to parse file-spec string and put expanded string
                        0101   154 ; into dynamic memory buffer
                        0101   155 ;
                        0101   156 ; Inputs:
                        0101   157 ;       R3 = NAM block address
                        0101   158 ;       R4 = FAB block address
                        0101   159 ;
                        0101   160 ; Outputs:
                        0101   161 ;       R0 = Success/error status
                        0101   162 ;
                        0101   163 ;
                        0101   164 PARSE_SPEC:
                    14  BB      0101   165          PUSHR   #^M<R2,R4>
                    47  10      0103   166          BSB     GET_FS_NODE             ; Get file-spec node
                41  50  E9      0105   167          BLBC    R0,20$                  ; Error if LBC
                        0108   168          $NAM_STORE NAM = R3, -
                        0108   169                  ESA = @W^VIRT_ADDR, ESS = #255
                        0113   170          $PARSE  FAB = R4                        ; Parse file name string
                24  50  E8      011C   171          BLBS    R0,10$                  ; OK if LBS
                0C  A4  DD      011F   172          PUSHL   FAB$L_STV(R4)           ; Signal error
                    50  DD      0122   173          PUSHL   R0
                        0124   174          ERRMSG  PRSERR,<R6,R7>
00000000'GF         02  FB      013A   175          CALLS   #2,G^LIB$SIGNAL
                    06  11      0141   176          BRB     20$
                        0143   177 10$:
            52  0B  A3  9A      0143   178          MOVZBL  NAM$B_ESL(R3),R2        ; Get expanded string size
                1F  10      0147   179          BSB     RETURN_FS_NODE          ; Return unused part of node
                        0149   180 20$:
                    14  BA      0149   181          POPR    #^M<R2,R4>
                        05  014B   182          RSB
```

```
                                014C   184          .SBTTL  GET_FS_NODE, RETURN_FS_NODE
                                014C   185 ;
                                014C   186 ; Subroutines to get and return file specification node
                                014C   187 ;
                                014C   188 ; Get node
                                014C   189 ;
                                014C   190 ; Inputs:
                                014C   191 ;       None
                                014C   192 ;
                                014C   193 ; Outputs:
                                014C   194 ;       R0 = Success/error status
                                014C   195 ;       VIRT_ADDR = Address of block
                                014C   196 ;       FILE_SIZE = Size of block
                                014C   197 ;
                                014C   198          .ENABL  LSB
                                014C   199 ;
                                014C   200 GET_FS_NODE:
0000'CF   00000100 8F   D0      014C   201          MOVL    #256,W^FILE_SIZE        ; Set size of expanded string buffer
                0000'CF   DF    0155   202          PUSHAL  W^VIRT_ADDR             ; Push parameters
                0000'CF   DF    0159   203          PUSHAL  W^FILE_SIZE
          00000000'GF    02     FB     015D   204          CALLS   #2,G^LIB$GET_VM
                   25 50   E9   0164   205          BLBC    R0,10$                  ; Error if LBC
                           05   0167   206          RSB
                                0168   207 ;
                                0168   208 ;
                                0168   209 ; Return node
                                0168   210 ;
                                0168   211 ; Inputs:
                                0168   212 ;       R2 = Number of bytes in node used
                                0168   213 ;       VIRT_ADDR = Address of node
                                0168   214 ;       FILE_SIZE = Size of node
                                0168   215 ;
                                0168   216 ; Outputs:
                                0169   217 ;       R0 = Success/error status
                                0168   218 ;       VIRT_ADDR = Address of memory returned
                                0168   219 ;       FILE_SIZE = Size of mempory returned
                                0168   220 ;
                                0168   221 ;
                                0168   222 RETURN_FS_NODE:
              52    07    C0    0168   223          ADDL2   #7,R2                   ; Round up to quadword
              52    07    CA    016B   224          BICL2   #7,R2
      0000'CF     52    C2    016E   225          SUBL2   R2,W^FILE_SIZE          ; Compute number of bytes to return
                  20    13    0173   226          BEQL    20$                     ; None if EQL
      0000'CF     52    C0    0175   227          ADDL2   R2,W^VIRT_ADDR          ; Address of bytes to return
              0000'CF   DF    017A   228          PUSHAL  W^VIRT_ADDR             ; Push parameters
              0000'CF   DF    017E   229          PUSHAL  W^FILE_SIZE
  00000000'GF     02    FB    0182   230          CALLS   #2,G^LIB$FREE_VM
                09 50   E8    0189   231          BLBS    R0,20$                  ; OK if LBS
                                018C   232 10$:
                   50   DD    018C   233          PUSHL   R0                      ; Signal error
  00000000'GF      01    FB    018E   234          CALLS   #1,G^LIB$SIGNAL
                                0195   235 20$:
                        05    0195   236          RSB
                                0196   237 ;
                                0196   238          .DSABL  LSB
```

```
                     0196   240          .SBTTL  OUTPUT_FILE
                     0196   241 ;
                     0196   242 ;
                     0196   243 ; Subroutine to get output file
                     0196   244 ;
                     0196   245 OUTPUT_FILE:
      56 0000'CF  DE 0196   246          MOVAL   W^INPUT_BUF,R6          ; Get address to put file name string
         0058     30 019B   247          BSBW    GETFILE                 ; Get next file
           57     D5 019E   248          TSTL    R7                      ; Is file spec null (0 bytes)
           35     13 01A0   249          BEQL    20$                     ; Yes if EQL
        1E 58     E9 01A2   250          BLBC    R8,10$                  ; Error if not last file
0000'CF 0000'8F   AB 01A5   251          BISW    #MERM_OUTPUT,W^MERGE_FLAGS ; Flag output file specified
      53 0000'CF  DE 01AC   252          MOVAL   W^OUTPUT_NAM,R3         ; Set NAM and FAB addresses
      54 0000'CF  DE 01B1   253          MOVAL   W^OUTPUT_FAB,R4
                     01B6   254          $FAB_STORE FAB = R4, FNA = (R6), FNS = R7
         FF40     30 01BE   255          BSBW    PARSE_SPEC
           32     11 01C1   256          BRB     40$
                     01C3   257 10$:
                     01C3   258          ERRMSG  ONEOUT
           1E     11 01D5   259          BRB     40$
                     01D7   260 20$:
        09 58     E9 01D7   261          BLBC    R8,30$                  ; Not at end of line if LBC
0000'CF 0000'8F   AA 01DA   262          BICW    #MERM_OUTPUT,W^MERGE_FLAGS ; Flag no output file
           12     11 01E1   263          BRB     40$
                     01E3   264 30$:
                     01E3   265          ERRMSG  NULLFS                  ; Report error
                     01F5   266 40$:
           05        01F5   267          RSB
```

```
                     01F6   269          .SBTTL  GETFILE
                     01F6   270 ;
                     01F6   271 ;
                     01F6   272 ; Subroutine to get next file spec from command line
                     01F6   273 ;
                     01F6   274 ; Inputs:
                     01F6   275 ;       R6  = Address to put file spec string
                     01F6   276 ;
                     01F6   277 ; Outputs:
                     01F6   278 ;       R0  = Success/error status
                     01F6   279 ;       R6  = Address of file-spec
                     01F6   280 ;       R7  = Size in bytes of file-spec
                     01F6   281 ;       R8  = Continue/terminate flag
                     01F6   282 ;
                     01F6   283 GETFILE:
        0040 8F BB   01F6   284          PUSHR   #^M<R6>
             57 D4   01FA   285          CLRL    R7                      ; file-spec sting
             53 D4   01FC   286          CLRL    R3                      ; Initialise [..] flag
                     01FE   287 10$:
             7B 10   01FE   288          BSB     GETCHAR                 ; Get next character
          6C 50 E9   0200   289          BLBC    R0,150$                 ; Error if LBC
             60 13   0203   290          BEQL    120$                    ; End of line if EQL
0274'CF   02 55 3A   0205   291          LOCC    R5,#2,W^LOCCHAR         ; Space or tab?
             F1 12   020B   292          BNEQ    10$                     ; Yes if NEQ
             07 11   020D   293          BRB     30$
                     020F   294 20$:
             6A 10   020F   295          BSB     GETCHAR                 ; Get next character
          5B 50 E9   0211   296          BLBC    R0,150$                 ; Error if LBC
             4F 13   0214   297          BEQL    120$                    ; End of line if EQL
                     0216   298 30$:
0274'CF   07 55 3A   0216   299          LOCC    R5,#7,W^LOCCHAR         ; Special character
      07  00 50 8F  021C   300          CASEB   R0,#0,#7
                001D' 0220   301 40$:     .WORD   80$-40$                 ; Normal character
                0010' 0222   302          .WORD   50$-40$                 ; >
                0014' 0224   303          .WORD   60$-40$                 ; <
                0010' 0226   304          .WORD   50$-40$                 ; ]
                0014' 0228   305          .WORD   60$-40$                 ; [
                0019' 022A   306          .WORD   70$-40$                 ; ,
                0024' 022C   307          .WORD   90$-40$                 ; Space
                0024' 022E   308          .WORD   90$-40$                 ; Tab
                     0230   309 50$:
             53 D4   0230   310          CLRL    R3                      ; Clear [..] flag
             09 11   0232   311          BRB     80$
                     0234   312 60$:
          53 01 D0   0234   313          MOVL    #1,R3                   ; Set [..] flag
             04 11   0237   314          BRB     80$
                     0239   315 70$:
       2D 53 00 E5   0239   316          BBCC    #0,R3,130$              ; If ',' but in [..] process as normal
                     023D   317 80$:
          86 55 90   023D   318          MOVB    R5,(R6)+                ; Copy byte to file-spec string
             57 D6   0240   319          INCL    R7                      ; and increment size
             CB 11   0242   320          BRB     20$                     ; Back for next character
                     0244   321 90$:
             35 10   0244   322          BSB     GETCHAR                 ; Get next character
          26 50 E9   0246   323          BLBC    R0,150$                 ; Error if LBC
             1A 13   0249   324          BEQL    120$                    ; End of line if EQL
0274'CF   03 55 3A   024B   325          LOCC    R5,#3,W^LOCCHAR         ; Trailing character?
```

```
                03   00   50    8F  0251    326                 CASEB   R0,#0,#3
                                0008' 0255    327 100$:         .WORD   110$-100$               ; No
                                0015' 0257    328               .WORD   130$-100$               ;
                                FFEF  0259    329               .WORD   90$-100$                ; $pace
                                FFEF  025B    330               .WORD   90$-100$                ; Tab
                                      025D    331 110$:
                      0000'CF   D7    025D    332               DECL    W^CMD_INPUT_POS         ; Back-up line pointer
                      0000'CF   D6    0261    333               INCL    W^CMD_INPUT_SZE
                                      0265    334 120$:
                      58   01   D0    0265    335               MOVL    #1,R8                   ; Set for no more input files
                           02   11    0268    336               BRB     140$
                                      026A    337 130$:
                           58   D4    026A    338               CLRL    R8                      ; Set for more input files
                                      026C    339 140$:
                      50   01   D0    026C    340               MOVL    #1,R0
                                      026F    341 150$:
                      0040 8F   BA    026F    342               POPR    #^M<R6>
                                05    0273    343               RSB
                                      0274    344 ;
3E 3C 5D 5B 2C 20 09                  0274    345 LOCCHAR:      .ASCII  <^X9>/ ,[]<>/
```

```
                     027B   347          .SBTTL  GETCHAR
                     027B   348 ;
                     027B   349 ;
                     027B   350 ; Subroutine to get next character from command line
                     027B   351 ;
                     027B   352 ; Inputs:
                     027B   353 ;       None
                     027B   354 ;
                     027B   355 ; Outputs:
                     027B   356 ;       R0  = Success/error status
                     027B   357 ;       R5  = character
                     027B   358 ;       'Z' = 0 if end of line
                     027B   359 ;       'Z' = 1 if valid character in R5
                     027B   360 ;
                     027B   361 ;
                     027B   362 GETCHAR:
        0300 8F   BB 027B   363          PUSHR   #^M<R8,R9>
        50   01   D0 027F   364          MOVL    #1,R0                   ; Assume success
    59  0000'CF   D0 0282   365          MOVL    W^CMD_INPUT_SZE,R9      ; Set command size
    58  0000'CF   D0 0287   366          MOVL    W^CMD_INPUT_POS,R8      ; Set command input position
            29    12 028C   367          BNEQ    30$                     ; Have a command line if NEQ
                     028E   368 10$:
                     028E   369          $GET    RAB = CMD_INPUT_RAB     ; Prompt for and get next command line
          0F 50   E8 029B   370          BLBS    R0,20$                  ; OK if LBS
        000C'CF   DD 029E   371          PUSHL   W^CMD_INPUT_RAB+RAB$L_STV ; Signal error
            50    DD 02A2   372          PUSHL   R0
00000000'GF     02    FB 02A4   373          CALLS   #2,G^LIB$SIGNAL
            50    11 02AB   374          BRB     70$
                     02AD   375 20$:
    58  0028'CF   D0 02AD   376          MOVL    W^CMD_INPUT_RAB+RAB$L_RBF,R8 ; Reset command line position
    59  0022'CF   3C 02B2   377          MOVZWL  W^CMD_INPUT_RAB+RAB$W_RSZ,R9 ; and size
                     02B7   378 30$:
            59    D5 02B7   379          TSTL    R9                      ; Any characters in line?
            1E    13 02B9   380          BEQL    40$                     ; No if EQL
        55  BB    90 02BB   381          MOVB    (R8)+,R5                ; Get character
            59    D7 02BE   382          DECL    R9                      ; Decrement character count
        2D  55    91 02C0   383          CMPB    R5,#^A/-/               ; Continuation character?
            2E    12 02C3   384          BNEQ    60$                     ; No if not equal
            59    D5 02C5   385          TSTL    R9                      ; Last character on line?
            16    12 02C7   386          BNEQ    50$                     ; No if NEQ
0030'CF 0001'CF   DE 02C9   387          MOVAL   W^PROMPT_CONT+1, -      ; Set continuation prompt
                     02D0   388                  W^CMD_INPUT_RAB+RAB$L_PBF
0034'CF 0000'CF   90 02D0   389          MOVB    W^PROMPT_CONT, -
                     02D7   390                  W^CMD_INPUT_RAB+RAB$B_PSZ
            B5    11 02D7   391          BRB     10$
                     02D9   392 40$:
            55    D4 02D9   393          CLRL    R5                      ; Clear character
            58    D4 02DB   394          CLRL    R8                      ; Clear valid command line flag
            14    11 02DD   395          BRB     60$
                     02DF   396 50$:
                     02DF   397          ERRMSG  INVPMD                  ; Issue error message
            0A    11 02F1   398          BRB     70$
                     02F3   399 60$:
    0000'CF   58  D0 02F3   400          MOVL    R8,W^CMD_INPUT_POS      ; Save command position
    0000'CF   59  D0 02F8   401          MOVL    R9,W^CMD_INPUT_SZE      ; and size
                     02FD   402 70$:
            55    D5 02FD   403          TSTL    R5                      ; Set condition codes
```

```
0300 8F   BA  02FF   404          POPR    #^M<R8,R9>
          05  0303   405          RSB
```

```
                     0304   407          .SBTTL  OPEN_FILES
                     0304   408 ;
                     0304   409 ;
                     0304   410 ; Procedure to open slipr input and output files
                     0304   411 ;
                     0304   412 ; Inputs:
                     0304   413 ;       R11 = number of input files
                     0304   414 ;
                     0304   415 ; Outputs:
                     0304   416 ;       None
                     0304   417 ;
                     0304   418 ;
                     0304   419 OPEN_FILES::
                0000 0304   420          .WORD   0
          5A 0000'CF  DE     0306   421          MOVAL   W^FILE_NODES,R10        ; Initialise file nodes pointer
                     030B   422 10$:
             5A  6A  D0     030B   423          MOVL    (R10),R10               ; Get next node
   00000000'8F  5A  D1     030E   424          CMPL    R10,#FILE_NODES         ; At end of list?
               07  13     0315   425          BEQL    20$                     ; Yes if EQL
               1B  10     0317   426          BSB     OPEN_INPUT              ; Open input file
            EF 50  E8     0319   427          BLBS    R0,10$                  ; OK if LBC
               15  11     031C   428          BRB     30$
                     031E   429 20$:
0004'CF  01000000 8F  C8     031E   430          BISL    #FAB$M_NAM,W^INPUT_FAB+FAB$L_FOP
    0000'CF  0000'8F  B3     0327   431          BITW    #MERM_OUTPUT,W^MERGE_FLAGS ; Was output file specified?
               03  13     032E   432          BEQL    30$                     ; No if EQL
             0099  30     0330   433          BSBW    CREATE_OUTPUT           ; Create output file
                     0333   434 30$:
               04    0333   435          RET
```

```
                                0334   437          .SBTTL  OPEN_INPUT
                                0334   438 ;
                                0334   439 ; Subroutine to open input file
                                0334   440 ;
                                0334   441 ; Inputs:
                                0334   442 ;       R10 = File node address
                                0334   443 ;
                                0334   444 ; Outputs:
                                0334   445 ;       R0 = Success/error code
                                0334   446 ;
                                0334   447 ;
                                0334   448 OPEN_INPUT:
                     53    5A   D0   0334   449          MOVL    R10,R3                  ; Set NAM block address
          53  00000048  8F   C0   0337   450          ADDL    #SLP$T_NAM,R3
               54  0000'CF   DE   033E   451          MOVAL   W^INPUT_FAB,R4          ; and FAB address
                      FE06   30   0343   452          BSBW    GET_FS_NODE             ; Get node for resultant file spec
                     7E 50   E9   0346   453          BLBC    R0,30$                  ; Error if LBC
                                0349   454          $FAB_STORE FAB = R4, NAM = (R3), -
                                0349   455                  FNA = @NAM$L_ESA(R3), FNS = NAM$B_ESL(R3)
                                0357   456          $NAM_STORE NAM = R3, ESS = #0, -
                                0357   457                  RSA = @VIRT_ADDR, RSS = #255
                                0367   458          $OPEN   FAB = R4                ; Open input file
                     29 50   E9   0370   459          BLBC    R0,20$                  ; Error if LBC
                                0373   460          $CLOSE  FAB = R4                ; Close file to release FAB
                     1D 50   E9   037C   461          BLBC    R0,20$                  ; Error if LBC
               52    03 A3   9A   037F   462          MOVZBL  NAM$B_RSL(R3),R2        ; Get number of bytes used
                      FDE2   30   0383   463          BSBW    RETURN_FS_NODE          ; and return rest of node
                     3E 50   E9   0386   464          BLBC    R0,30$                  ; Error if LBC
                         52   D4   0389   465          CLRL    R2                      ; Return Expanded fs node
          0000'CF    2C A4   D0   038B   466          MOVL    FAB$L_FNA(R4),W^VIRT_ADDR
          0000'CF    34 A4   9A   0391   467          MOVZBL  FAB$B_FNS(R4),W^FILE_SIZE
                       FDCE   30   0397   468          BSBW    RETURN_FS_NODE
                         2B   11   039A   469          BRB     30$
                              039C   470 20$:
               56    2C A4   D0   039C   471          MOVL    FAB$L_FNA(R4),R6        ; Get file spec
               57    34 A4   9A   03A0   472          MOVZBL  FAB$B_FNS(R4),R7
                              03A4   473          ERRMSG  OPENER,<R6,R7>
                     0C A4   DD   03BA   474          PUSHL   FAB$L_STV(R4)           ; Signal error
                     08 A4   DD   03BD   475          PUSHL   FAB$L_STS(R4)
     00000000'GF       02   FB   03C0   476          CALLS   #2,G^LIB$SIGNAL
                              03C7   477 30$:
                000C CA   94   03C7   478          CLRB    W^SLP$B_FLAGS(R10)      ; Initialise flags
                         05   03CB   479          RSB
```

```
                     03CC    481          .SBTTL  CREATE_OUTPUT
                     03CC    482 ;
                     03CC    483 ; Subroutine to create output file
                     03CC    484 ;
                     03CC    485 ; Inputs:
                     03CC    486 ;       None
                     03CC    487 ;
                     03CC    488 ; Outputs:
                     03CC    489 ;       R0 = Success/error status
                     03CC    490 ;
                     03CC    491 ;
                     03CC    492 CREATE_OUTPUT:
53   0000'CF   DE    03CC    493          MOVAL   W^OUTPUT_NAM,R3          ; Set NAM and
54   0000'CF   DE    03D1    494          MOVAL   W^OUTPUT_FAB,R4          ; FAB pointers
                     03D6    495          $FAB_STORE FAB = R4, -
                     03D6    496                  FNA = @NAM$L_ESA(R3), FNS = NAM$B_ESL(R3)
         FD69    30  03E0    497          BSBW    GET_FS_NODE              ; Get file_spec node
       7C 50     E9  03E3    498          BLBC    R0,40$                   ; Error if LBC
                     03E6    499          $NAM_STORE NAM = R3, ESS = #0, -
                     03E6    500                  RSA = @VIRT_ADDR, RSS = #255
                     03F6    501          $CREATE FAB = R4                 ; Open output file
       05 50     E8  03FF    502          BLBS    R0,10$                   ; OK if LBS
       0C A4     DD  0402    503          PUSHL   FAB$L_STV(R4)            ; Signal error
          14     11  0405    504          BRB     20$
                     0407    505 10$:
                     0407    506          $CONNECT RAB = OUTPUT_RAB        ; Connect RAB to FAB
       30 50     E8  0414    507          BLBS    R0,30$                   ; OK if LBS
     000C'CF     DD  0417    508          PUSHL   W^OUTPUT_RAB+RAB$L_STV   ; Signal error
                     041B    509 20$:
          50     DD  041B    510          PUSHL   R0
       56   2C A4   D0  041D    511          MOVL    FAB$L_FNA(R4),R6         ; Get file spec
       57   34 A4   9A  0421    512          MOVZBL  FAB$B_FNS(R4),R7
                     0425    513          ERRMSG  CREATE,<R6,R7>
          50   6E   D0  043B    514          MOVL    (SP),R0                  ; Reset R0
00000000'GF    02   FB  043E    515          CALLS   #2,G^LIB$SIGNAL
               1B   11  0445    516          BRB     40$
                     0447    517 30$:
       52   03 A3   9A  0447    518          MOVZBL  NAM$B_RSL(R3),R2         ; Get number of bytes used
         FD1A      30  044B    519          BSBW    RETURN_FS_NODE           ; and return rest of node
       11 50     E9  044E    520          BLBC    R0,40$                   ; Error of LBC
          52     D4  0451    521          CLRL    R2                       ; Return expanded fs node
0000'CF   2C A4   D0  0453    522          MOVL    FAB$L_FNA(R4),W^VIRT_ADDR
0000'CF   34 A4   9A  0459    523          MOVZBL  FAB$B_FNS(R4),W^FILE_SIZE
         FD06     30  045F    524          BSBW    RETURN_FS_NODE
                     0462    525 40$:
                  05 0462    526          RSB
```

```
                     0463   528          .SBTTL  CLOSE_FILES
                     0463   529 ;
                     0463   530 ;
                     0463   531 ; Procedure to close files
                     0463   532 ;
                     0463   533 ; Inputs:
                     0463   534 ;       File list
                     0463   535 ;
                     0463   536 ; Outputs:
                     0463   537 ;       None
                     0463   538 ;
                     0463   539 ;
                     0463   540 CLOSE_FILES::
                0000 0463   541          .WORD   0
                     0465   542
52   0000'CF    DE   0465   543          MOVAL   W^INPUT_FAB,R2
          13    10   046A   544          BSB     CLOSE
52   0000'CF    DE   046C   545          MOVAL   W^OUTPUT_FAB,R2
          0C    10   0471   546          BSB     CLOSE
52   0000'CF    DE   0473   547          MOVAL   W^RANDOM_FAB,R2
          05    10   0478   548          BSB     CLOSE
     0000'CF    D4   047A   549          CLRL    W^RANDOM_FILE
          04    047E   550          RET
                     047F   551 ;
                     047F   552 ;
                     047F   553 ; Subroutine to close file
                     047F   554 ;
                     047F   555 ; Inputs:
                     047F   556 ;       R2 = FAB address
                     047F   557 ;
                     047F   558 ; Outputs:
                     047F   559 ;       None
                     047F   560 ;
                     047F   561 CLOSE:
02 A2    B5   047F   562          TSTW    FAB$W_IFI(R2)           ; Is file open?
     09    13   0482   563          BEQL    10$                     ; No if EQL
               0484   564          $CLOSE  FAB = R2                ; Yes it's open so close it
               048D   565 10$:
          05   048D   566          RSB
               048E   567
               048E   568 ;
               048E   569 ;
               048E   570          .END
```

```
$$                              =  00000002
$$.TMP1                         =  00000001
$$.TMP2                         =  00000052
..AFLG                          =  00000000
..FLG                           =  00000002
..MOD                           =  00000000
..TYP                           =  00000053
.LEN                            =  00000001
BUF_SIZE                        =  00000200
CLOSE                              0000047F  R      02
CLOSE_FILES                        00000463  RG     02
CMD_INPUT_POS                      ********    X    02
CMD_INPUT_RAB                      ********    X    02
CMD_INPUT_SZE                      ********    X    02
CMD_SIZE                        =  00000084
CREATE_OUTPUT                      000003CC  R      02
DEF_AUDIT                          ********    X    02
DEF_NAME                           ********    X    02
ED$B_FILENO                     =  00000019
ED$B_FLAGS                      =  00000018
ED$K_BLN                        =  0000001A
ED$L_BWD                        =  00000004
ED$L_FILE                       =  00000014
ED$L_FWD                        =  00000000
ED$W_LINES                      =  0000000C
ED$W_LOC1                       =  00000008
ED$W_LOC2                       =  0000000A
ED$W_RFA                        =  0000000E
FAB$B_DNS                       =  00000035
FAB$B_FNS                       =  00000034
FAB$L_DNA                       =  00000030
FAB$L_FNA                       =  0000002C
FAB$L_FOP                       =  00000004
FAB$L_NAM                       =  00000028
FAB$L_STS                       =  00000008
FAB$L_STV                       =  0000000C
FAB$M_NAM                       =  01000000
FAB$W_IFI                       =  00000002
FILE_NODES                         ********    X    02
FILE_SIZE                          ********    X    02
GETCHAR                            0000027B  R      02
GETFILE                            000001F6  R      02
GET_FILES                          00000000  RG     02
GET_FS_NODE                        0000014C  R      02
GET_HANDLER                        ********    X    02
INPUT_BUF                          ********    X    02
INPUT_FAB                          ********    X    02
INPUT_FILES                        00000033  R      02
INPUT_SPEC                         00000078  R      02
LIB$FREE_VM                        ********    X    02
LIB$GET_VM                         ********    X    02
LIB$SIGNAL                         ********    X    02
LOCCHAR                            00000274  R      02
MER$_CREATE                        ********    X    02
MER$_INVPMD                        ********    X    02
MER$_NULLFS                        ********    X    02
MER$_ONEOUT                        ********    X    02
MER$_OPENER                        ********    X    02
MER$_PRSERR                        ********    X    02
MERGE_FLAGS                        ********    X    02
MERM_OUTPUT                        ********    X    02
NAM$B_BID                       =  00000000
NAM$B_BLN                       =  00000001
NAM$B_ESL                       =  0000000B
NAM$B_ESS                       =  0000000A
NAM$B_RSL                       =  00000003
NAM$B_RSS                       =  00000002
NAM$C_BID                       =  00000002
NAM$C_BLN                       =  00000060
NAM$K_BLN                       =  00000060
NAM$L_ESA                       =  0000000C
NAM$L_RSA                       =  00000004
OPEN_FILES                         00000304  RG     02
OPEN_INPUT                         00000334  R      02
OUTPUT_FAB                         ********    X    02
OUTPUT_FILE                        00000196  R      02
OUTPUT_NAM                         ********    X    02
OUTPUT_RAB                         ********    X    02
PARSE_SPEC                         00000101  R      02
PROMPT_CONT                        ********    X    02
PROMPT_INPUT                       ********    X    02
PROMPT_OUTPUT                      ********    X    02
RAB$B_PSZ                       =  00000034
RAB$L_PBF                       =  00000030
RAB$L_RBF                       =  00000028
RAB$L_STV                       =  0000000C
RAB$W_RSZ                       =  00000022
RANDOM_FAB                         ********    X    02
RANDOM_FILE                        ********    X    02
RETURN_FS_NODE                     00000168  R      02
SLP$B_FILENO                    =  0000000D
SLP$B_FLAGS                     =  0000000C
SLP$K_BLN                       =  000000A8
SLP$L_BWD                       =  00000004
SLP$L_FWD                       =  00000000
SLP$Q_AUCDS                     =  00000028
SLP$Q_AUDDS                     =  00000010
SLP$Q_CMNT                      =  00000040
SLP$T_AUCST                     =  00000030
SLP$T_AUDST                     =  00000018
SLP$T_NAM                       =  00000048
SLP$W_DOT                       =  0000000E
SLP$W_LOC1                      =  00000008
SLP$W_LOC2                      =  0000000A
SLP_SIZE                           ********     X    02
SYS$CLOSE                          ********    GX    02
SYS$CONNECT                        ********    GX    02
SYS$CREATE                         ********    GX    02
SYS$GET                            ********    GX    02
SYS$OPEN                           ********    GX    02
SYS$PARSE                          ********    GX    02
VIRT_ADDR                          ********     X    02
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$CODE	0000048E (1166.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.09	00:00:00.54
Command processing	110	00:00:00.71	00:00:01.61
Pass 1	304	00:00:11.27	00:00:16.98
Symbol table sort	0	00:00:00.94	00:00:01.01
Pass 2	119	00:00:02.46	00:00:03.61
Symbol table output	14	00:00:00.09	00:00:00.09
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	582	00:00:15.60	00:00:23.90

The working set limit was 1200 pages.
57645 bytes (113 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 731 non-local and 48 local symbols.
655 source lines were read in Pass 1, producing 19 object records in Pass 2.
38 pages of virtual memory were used to define 27 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	23
TOTALS (all libraries)	23

969 GETS were required to define 23 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SUMFILES/OBJ=OBJ$:SUMFILES MSRC$:SUMCOM/UPDATE=(ENH$:SUMCOM)+MSRC$:SUMFILES/UPDATE=(ENH$:SUMFILES)+EXECML$/LIB

SUMLIST
LIS

SUMMAIN
LIS

SUMTFRVEC
LIS

SUMFILES
LIS

SUMMSG
LIS

SUMERROR
LIS

SUMOPEN
LIS

SYS

UPDATE
LIS

SYS
MAP